

## **REMARKS**

### **Status of the Application**

Claims 1 through 18 are pending in the present application. Claim 11 is objected to for being a substantial duplicate of claim 10. Claims 9 and 18 have been objected to under 35 U.S.C. § 112, ¶ 2 as being indefinite. Claims 4, 5, 7, 9 through 14, 16, and 18 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Chandramohan et al. (U.S. Patent No. 6,711,619), in view of Morton ("Reading CGI Data: URL-encoding and the CGI protocol"). Claims 1-3, 6 and 15 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Chandramohan et al. in view of Morton, and in further view of Krintz et al. ("Reducing the overhead of dynamic compilation"). Claims 8 and 17 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Chandramohan et al., in view of Morton, and in further view of French (U.S. Patent No. 6,654,794). Claims 1, 4, 9, and 18 have been amended. No new matter has been added.

### **Claim Objections**

The Examiner has objected to claim 11 has been objected to under 37 C.F.R. § 1.75 as being a substantial duplicate of claim 10. Applicant respectfully submits that claim 10 has been incorrectly written by Applicants in the listing of the claims section beginning with the response filed April 27, 2004. Applicants apologize for the typographical error and have corrected claim 10 in the present listing of the claims. Applicants submit that claim 11 is not a duplicate of claim 10 as originally filed. It is therefore respectfully requested that the Examiner withdraw the objection to claim 11.

### **35 U.S.C. § 112 Rejections**

Claims 9 and 18 were rejected under 35 U.S.C. § 112 as being indefinite. While Applicants disagree that claims 9 and 18 are indefinite, for the purposes of advancing prosecution, Applicants have amended claims 9 and 18 to overcome the rejection. It is respectively requested that the Examiner withdraw the rejections and allow claims 9 and 18.

### Prior Art Rejections

#### *The Disclosed Systems and Methods*

Claim 1 as amended is directed to a computer-based method for compiling a source code file on a client computer, the source code file being stored on a remote server computer and being accessible via web protocols, the method comprising:

(a) accepting a manually specified compile command, the compile command including a set of parameters, the set of parameters including an identifier corresponding to the source code file; and

(b) executing by a compiler stored on the client computer a compile procedure corresponding to the compile command, the compile procedure effecting conversion of the source code file into a file executable on the client computer,

wherein step (b) includes downloading the source code file from the remote server computer to the client computer using web protocols without executing a manually specified download command, **and the conversion of the source code file into a file executable on the client computer by the compiler begins before the source code file has been completely downloaded to the client computer,** and

further wherein the identifier corresponding to the source code comprises an identifier of executable code, and downloading the source code file comprises transmitting to the remote server the identifier corresponding to executable code and at least one parameter used by the executable code to identify the source code.

Claim 4 as amended is directed to a computer-based method for executing an application on a client computer, the application functioning to process file data stored on a remote server computer, the file data stored on the remote server computer being accessible via web protocols, the method comprising:

(a) accepting a manually specified execute command for an application stored on the client computer, the execute command including a set of parameters, the set of parameters including an identifier corresponding to the file data; and

(b) executing a procedure corresponding to the execute command, the procedure manipulating the file data on the client computer by the application,

wherein step (b) includes downloading the file data from the remote server computer to the client computer using the web protocols without executing a manually specified download command, and **the manipulation of the file data on the client computer by the application begins before the file data has been completely downloaded to the client computer**, and

further wherein downloading the file data comprises transmitting to the remote server computer an identifier of executable code and at least one parameter used by the executable code to derive the file data.

Claim 9 as amended is directed to a computer system including a processor, memory associated with the processor, and a storage medium capable of storing a data file, the data file having a corresponding file identifier, the system comprising:

(a) an application software component comprised of instructions in the memory and executable by the processor, the application software component functioning to process the data file; and

(b) an I/O software component comprised of instructions in the memory and executable by the processor, the I/O software component functioning to accept the file identifier, to determine whether the file identifier is a URL and, if so, to retrieve the data file from a remote server using the file identifier and, if not, to retrieve the data file from the storage medium using the file identifier,

wherein **the application software component processing the data file begins before the data file has been completely retrieved from the remote server** if it is determined that the file identifier is a URL,

wherein said file identifier identifies executable code, and

wherein said I/O software component functioning to retrieve the data file from a remote server using the file identifier operates by transmitting to the remote server said file identifier with at least one parameter, said at least one parameter being executable by the executable code identified by said file identifier.

Claim 18 as amended is directed to a computer-readable storage medium used in a computer system having a processor, memory associated with the processor and a storage device having a data storage medium, the computer-readable storage medium having instructions capable of being executed by the processor for performing the following:

- (a) accepting a file identifier corresponding to a data file;
- (b) determining whether the file identifier is a URL and, if so, retrieving the data file from a remote server using the file identifier and, if not, retrieving the data file from the data storage medium using the file identifier; and

**an application executing at the processor manipulating the data file before the data file has been completely retrieved from the remote server** if it is determined that the file identifier is a URL,

wherein said file identifier identifies executable code and retrieving the data file from a remote server comprises transmitting the file identifier and at least one parameter for executing the executable code.

In order for a reference or set of references to render these independent claims obvious, the reference must disclose each of the claimed elements, including those emphasized. More particularly, the references must teach the claimed combinations including **the manipulation of the file data on the client computer by the application begins before the file data has been completely downloaded to the client computer.** Applicants' undersigned representative respectfully submits that none of the references even teach the emphasized limitations, and cannot possibly suggest their combination with the other claimed elements.

Chandramohan et al. purports to disclose a method for executing portions of applications stored on a network (Chandramohan et al., Abstract). The applications are streamed to clients for execution, allowing the clients to execute the applications without storing them locally (Id.). The applications are speculatively streamed in non-sequential

ordered chunks, thereby reducing network latency effects associated with the transmission of code chunks (Id.).

Morton purports to disclose a general overview of the CGI protocol (Morton, page 1). Morton includes a description of common methods for transmitting data to a CGI script, including the GET and POST methods, as well as using the query string and path info (Id.). In addition, Morton describes how a web browser packages CGI data using url-encoding (Id.).

Krintz et al. purports to teach a compilation technique using idle cycles in multiprocessor systems to overlap compilation with application execution (Krintz et al., Summary). Compilation is done in a separate thread from the application, to reduce the possibility of delay in execution (Id.). Profile information is used to prioritize methods for background compilation, such that performance-critical methods are invoked as soon as possible (Id.).

French purports to teach a data processing system that that permits a client system to access a remote resource at a server coupled to the client system by a data network (French, Abstract). An operating system receives file system requests from client devices. The operating system identifies the file system requests as corresponding to a remote resource, and routes the request to a file system driver (Id.). The driver then converts the request to one usable by the remote server, such as HTTP (Id.).

However, neither Chandramohan et al., Morton, Krintz et al., nor French alone or in combination, **teach the manipulation of the file data on the client computer by the application before the file data has been completely downloaded to the client computer**, as required by the independent claims. The Examiner states that Chandramohan et al. teaches “manipulating the file data on the client computer” at column 7, lines 33-41 (Office Action, page 4). To the contrary, the cited portion describes the e-server receiving an e-client request to begin streaming an application to the e-client. This is not the application manipulating the data file by the client computer as required by the claims.

The Examiner is confusing the claimed systems and methods with what is disclosed in Chandramohan et al. The claimed systems and methods are directed to incorporating into software applications executing on a client computer functionality allowing for the applications to manipulate both local and remote data files. If a data file is remote, the data

file is downloaded to the client automatically and the application may immediately begin manipulating the data file before it completes downloading.

In contrast, Chandramohan et al. is directed to allowing clients to locally execute applications stored on remote computer devices. In Chandramohan et al. it is the applications that are downloaded to the client device, not the data files. As stated above, there is no mention of data files anywhere in Chandramohan et al.

Similarly, neither Morton, Krintz et al. nor French, teach **the manipulation of the file data on the client computer by the application begins before the file data has been completely downloaded to the client computer**. Accordingly, the combination of Chandramohan et al., Morton, Krintz et al., and French cannot possibly render the independent claims invalid. It is therefore respectfully submitted that the Examiner withdraw the rejections and allow claims 1, 4, 9, and 18.

Dependent claims 2, 3, 5 through 8, and 10 through 17, are all variously dependent on independent claims 1, 4, and 9, and are therefore allowable for at least the reasons given above for the independent claims. It is therefore respectfully submitted that the Examiner withdraw the rejections and allow claims 2, 3, 5 through 8, and 10 through 17.

### Conclusion

Applicants' undersigned representative respectfully requests reconsideration of the claims and early issuance of a Notice of Allowance. If the Examiner does not find the application is in condition for allowance, Applicant's undersigned representative respectfully requests that the Examiner contact him at the telephone number listed below before taking further action.

**DOCKET NO.:** MSFT-0234/155631.1  
**Application No.:** 09/767,768  
**Office Action Dated:** May 18, 2005

**Date:** 7/18/05

**PATENT**



John E. McGlynn  
Registration No. 42,863

Woodcock Washburn LLP  
One Liberty Place - 46th Floor  
Philadelphia PA 19103  
Telephone: (215) 568-3100  
Facsimile: (215) 568-3439